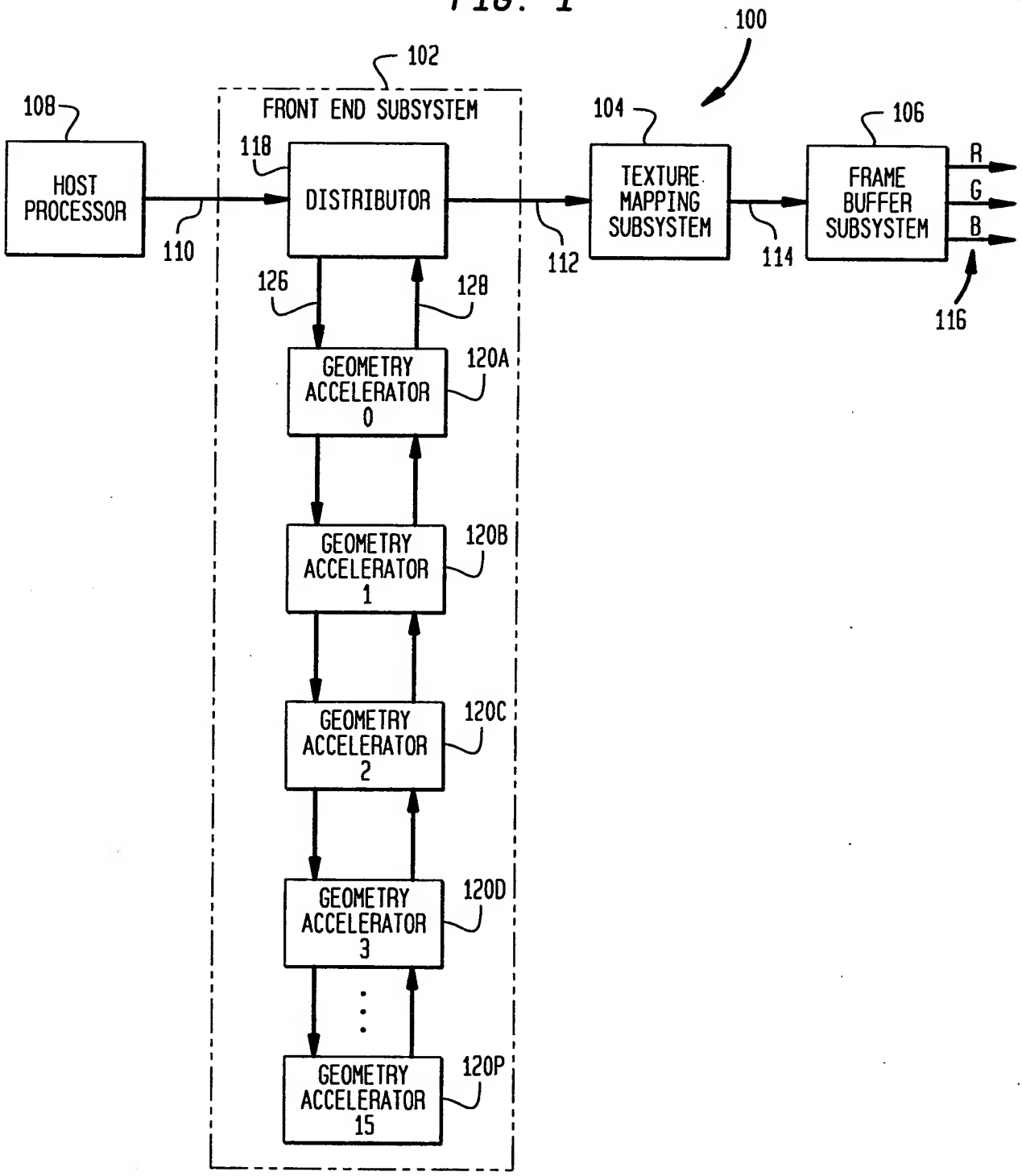
FIG. 1


100
102
FRONT END SUBSYSTEM
108
HOST PROCESSOR
110
118
DISTRIBUTOR
112
104
TEXTURE MAPPING SUBSYSTEM
114
106
FRAME BUFFER SUBSYSTEM
R
G
B
116
126
128
GEOMETRY ACCELERATOR 0
120A
GEOMETRY ACCELERATOR 1
120B
GEOMETRY ACCELERATOR 2
120C
GEOMETRY ACCELERATOR 3
120D
GEOMETRY ACCELERATOR 15
120P

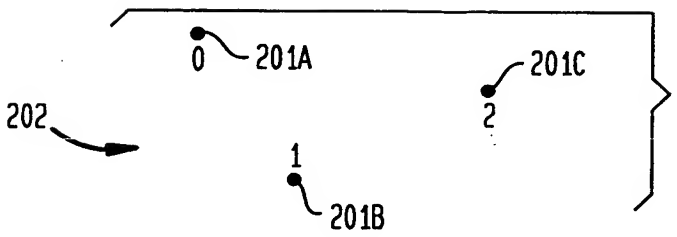


FIG. 2A

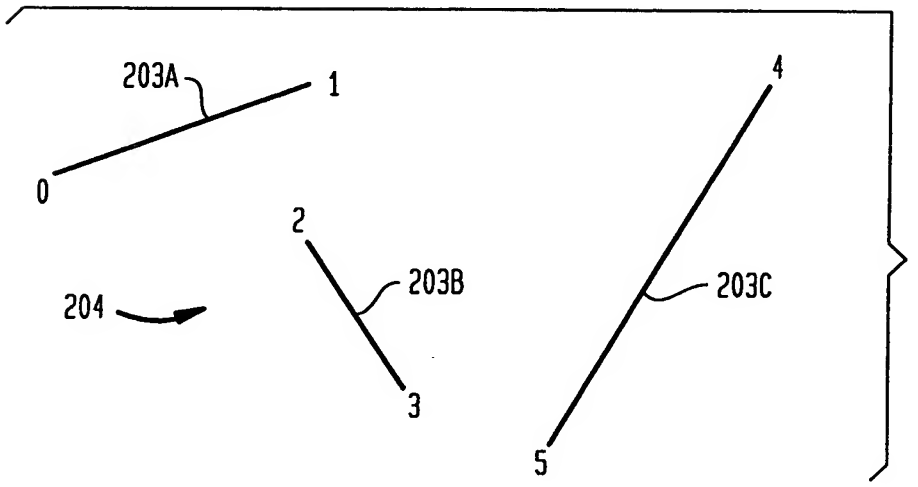


FIG. 2B

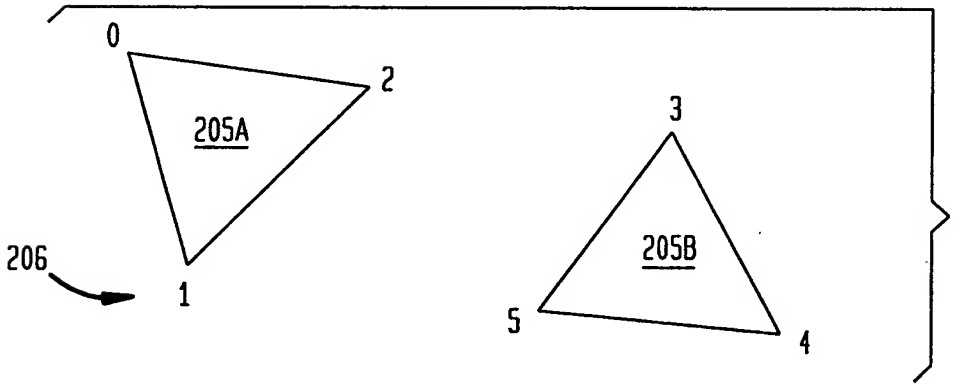


FIG. 2C

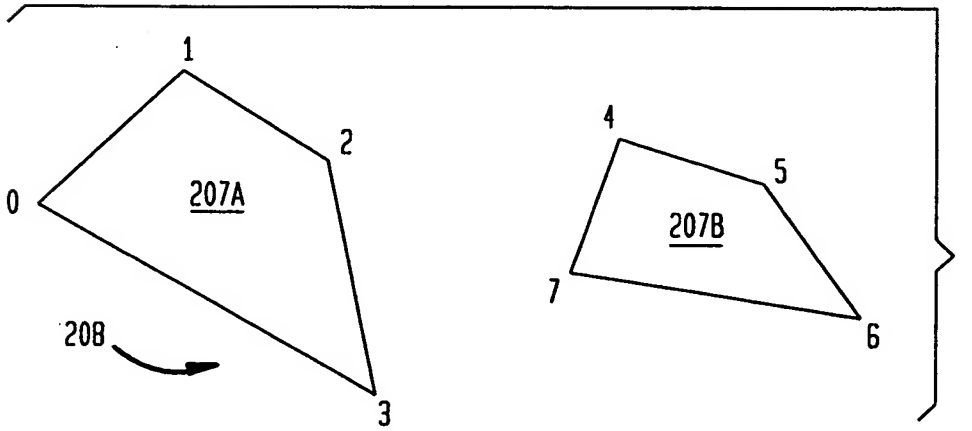


FIG. 2D

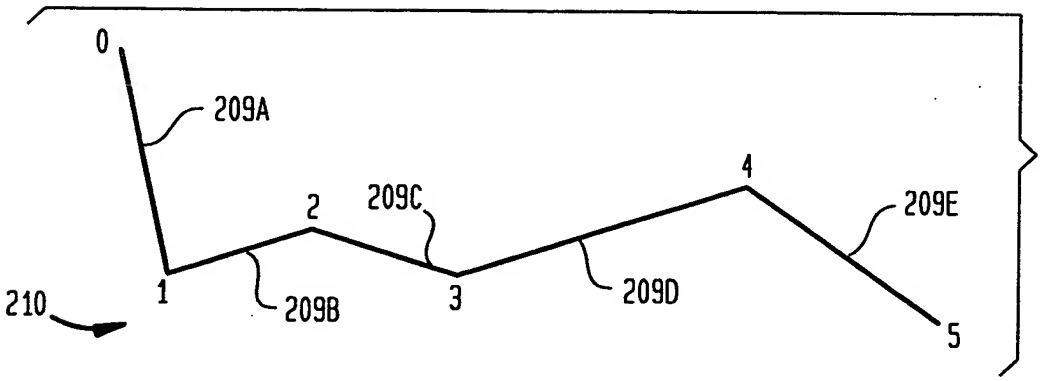


FIG. 2E

FIG. 2F

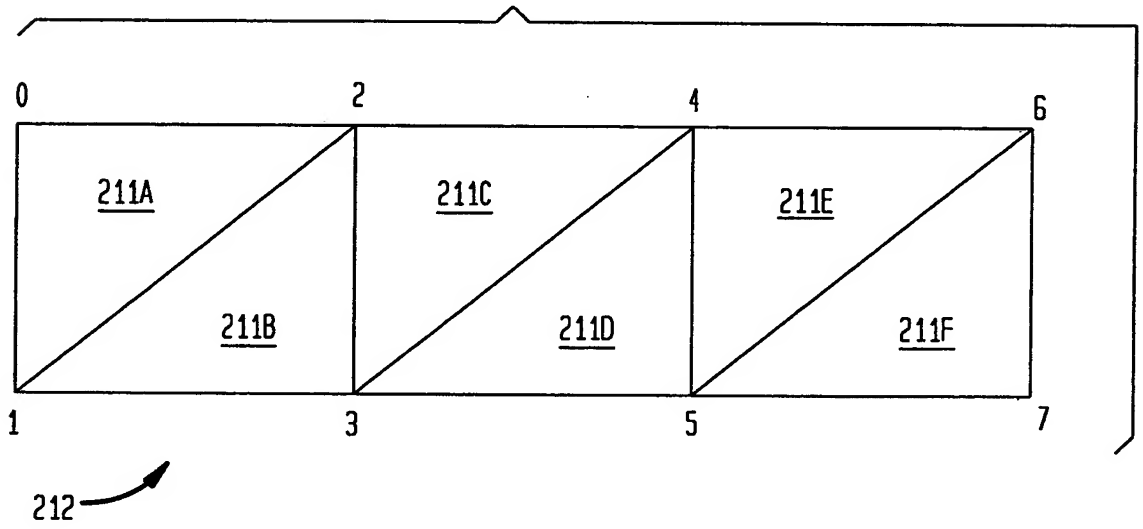


FIG. 2G

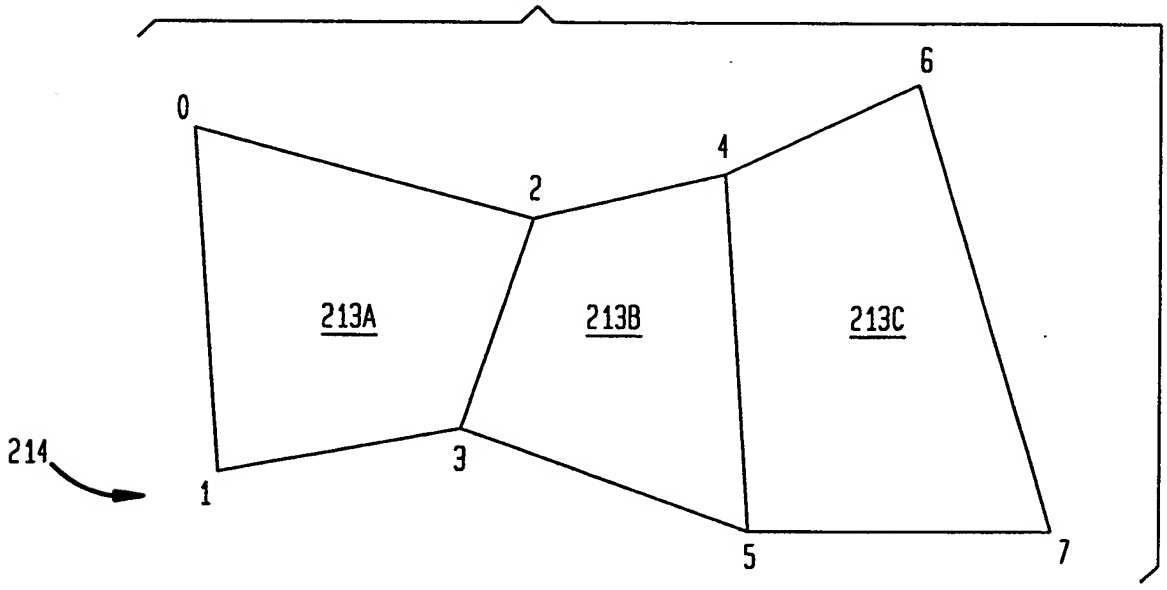


FIG. 2H

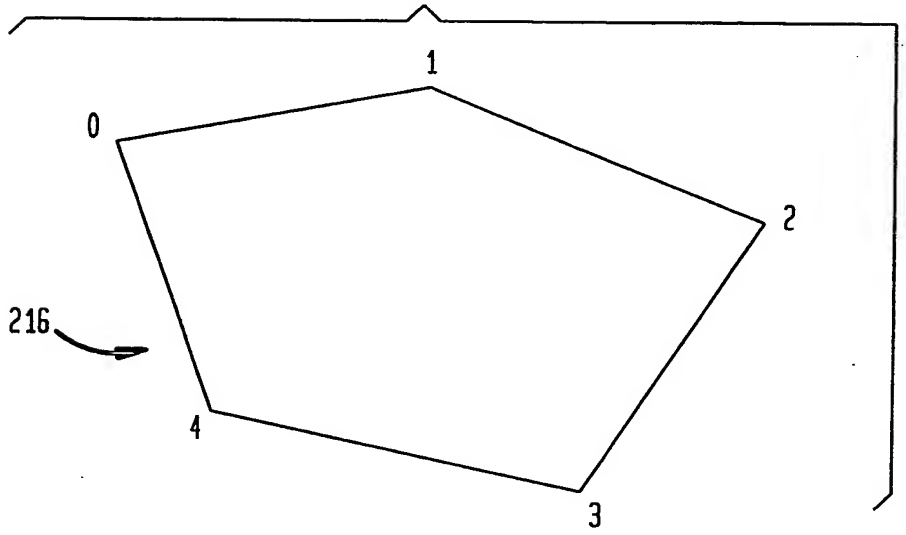


FIG. 2I

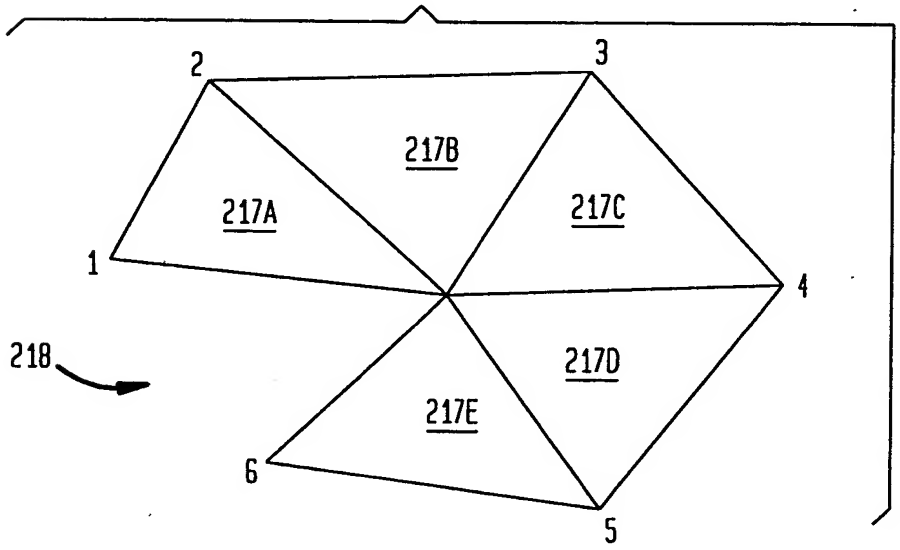


FIG. 2J

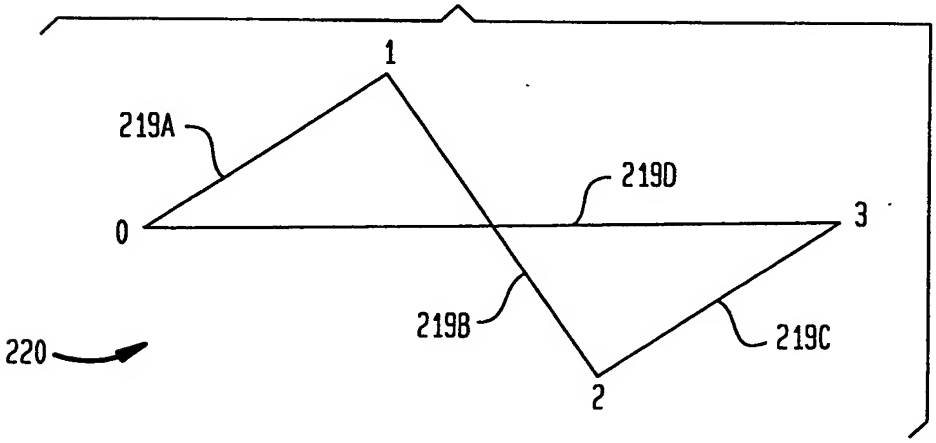


FIG. 3

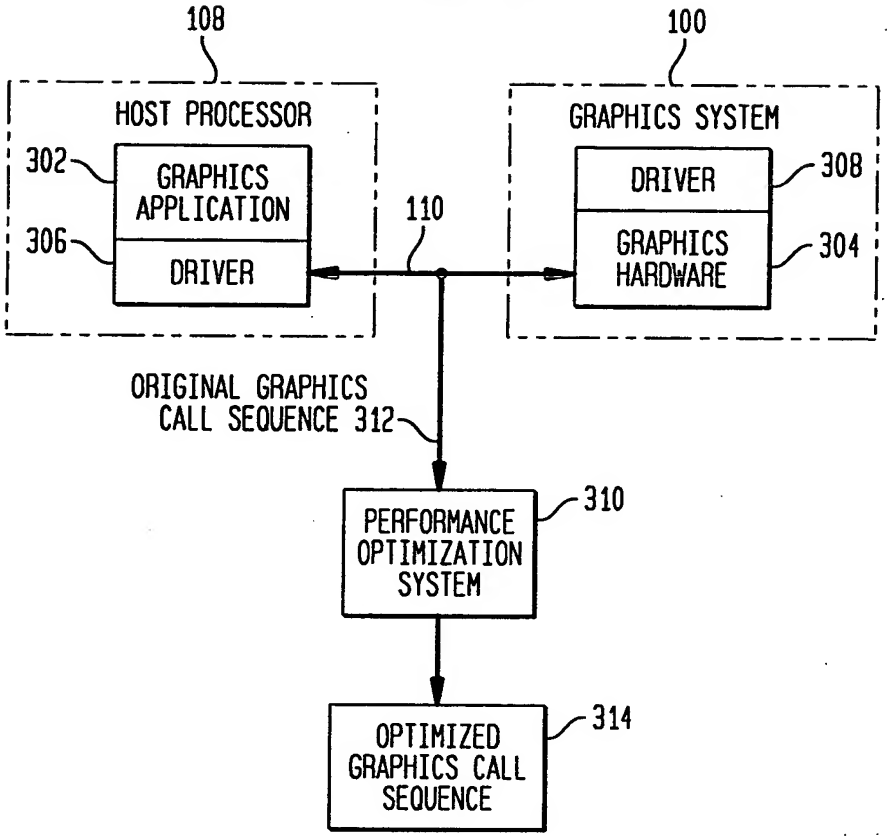

108
100
HOST PROCESSOR
GRAPHICS SYSTEM
302
GRAPHICS APPLICATION
306
DRIVER
110
DRIVER
308
GRAPHICS HARDWARE
304
ORIGINAL GRAPHICS CALL SEQUENCE 312
PERFORMANCE OPTIMIZATION SYSTEM
310
OPTIMIZED GRAPHICS CALL SEQUENCE
314

FIG. 4

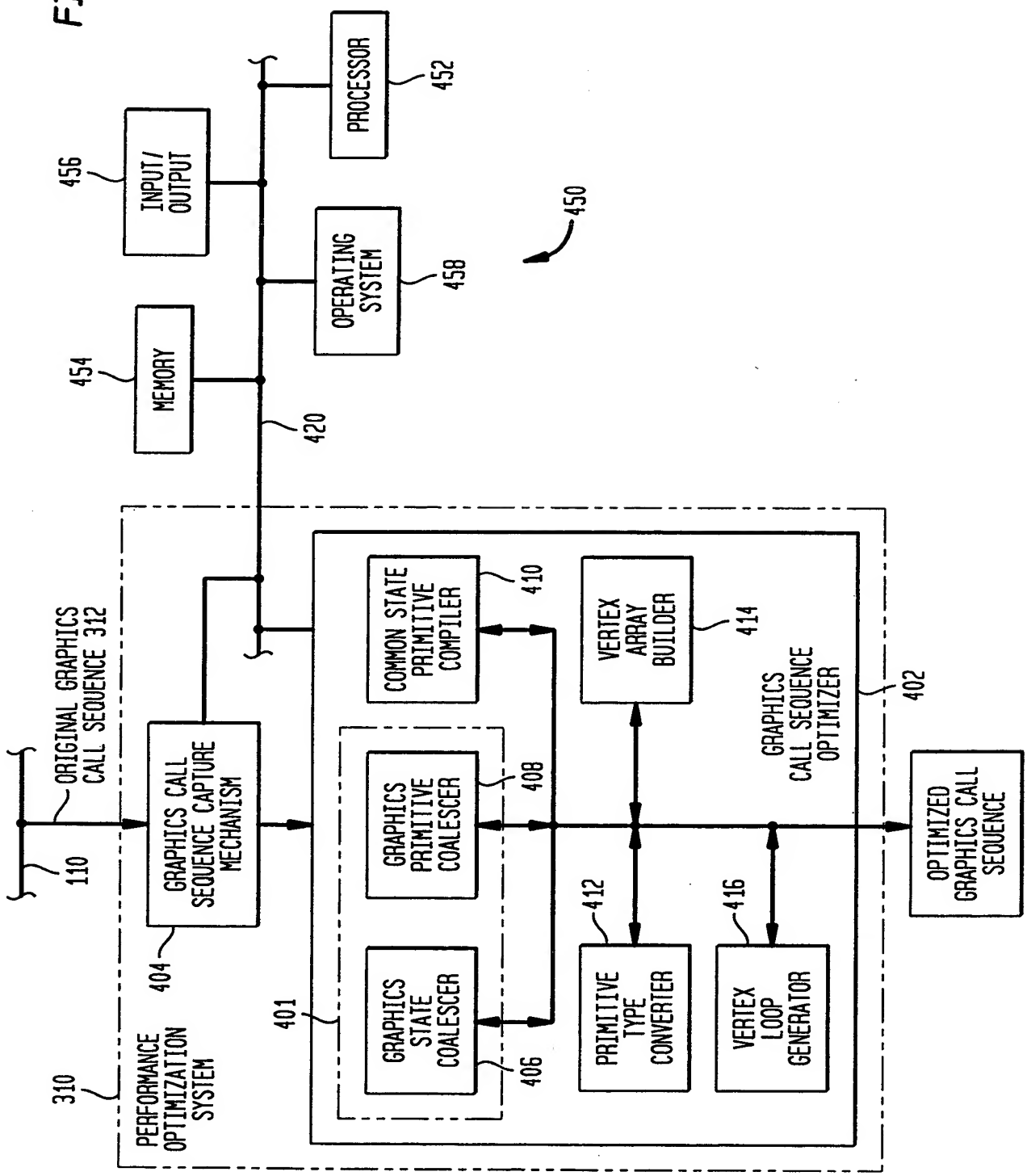

PROCESSOR
452
INPUT/
OUTPUT
456
OPERATING
SYSTEM
458
450
MEMORY
454
420
ORIGINAL GRAPHICS
CALL SEQUENCE 312
110
GRAPHICS CALL
SEQUENCE CAPTURE
MECHANISM
404
PERFORMANCE
OPTIMIZATION
SYSTEM
310
401
COMMON STATE
PRIMITIVE
COMPILER
410
GRAPHICS
PRIMITIVE
COALESCER
408
GRAPHICS
STATE
COALESCER
406
VERTEX
ARRAY
BUILDER
414
GRAPHICS
CALL SEQUENCE
OPTIMIZER
402
412
PRIMITIVE
TYPE
CONVERTER
416
VERTEX
LOOP
GENERATOR
OPTIMIZED
GRAPHICS CALL
SEQUENCE

FIG. 5A

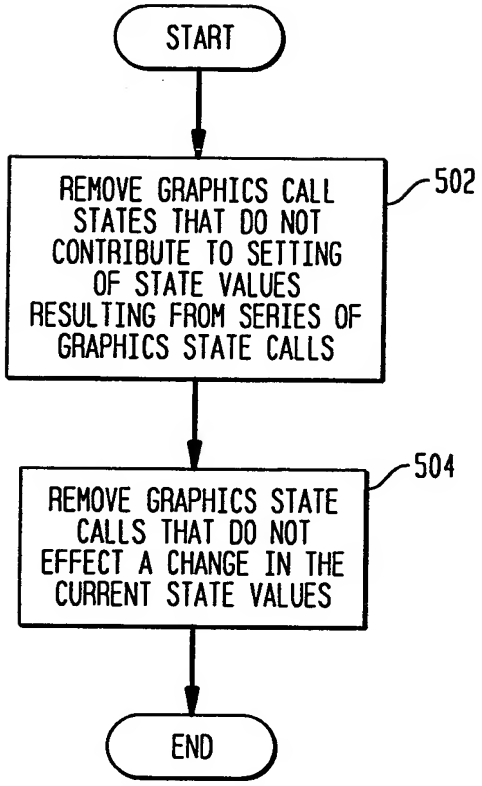

START
REMOVE GRAPHICS CALL STATES THAT DO NOT CONTRIBUTE TO SETTING OF STATE VALUES RESULTING FROM SERIES OF GRAPHICS STATE CALLS
502
REMOVE GRAPHICS STATE CALLS THAT DO NOT EFFECT A CHANGE IN THE CURRENT STATE VALUES
504
END
STATE COALESING PROCESS 500

FIG. 5B

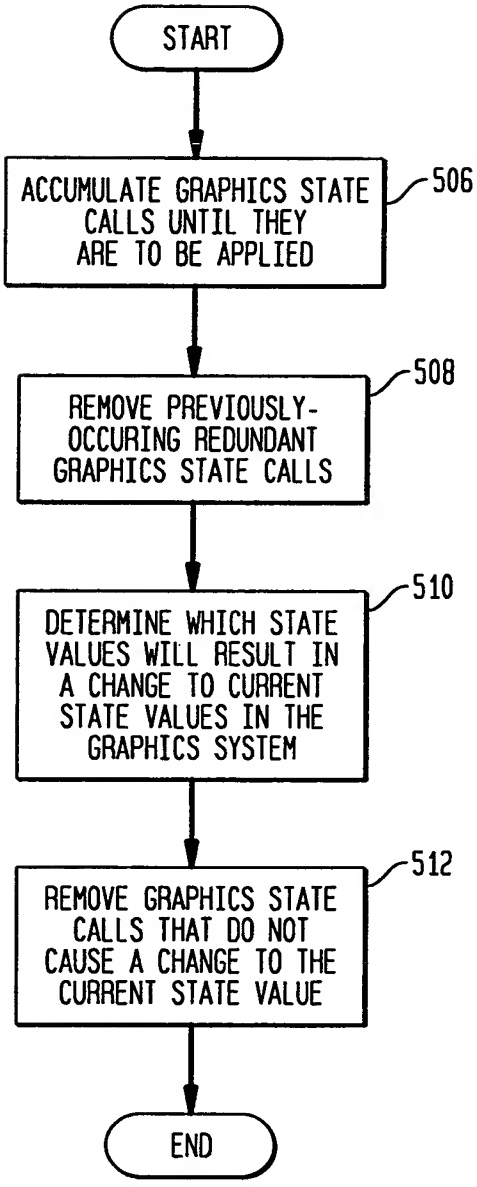

START
ACCUMULATE GRAPHICS STATE CALLS UNTIL THEY ARE TO BE APPLIED
506
REMOVE PREVIOUSLY-OCCURING REDUNDANT GRAPHICS STATE CALLS
508
DETERMINE WHICH STATE VALUES WILL RESULT IN A CHANGE TO CURRENT STATE VALUES IN THE GRAPHICS SYSTEM
510
REMOVE GRAPHICS STATE CALLS THAT DO NOT CAUSE A CHANGE TO THE CURRENT STATE VALUE
512
END
500

FIG. 5C

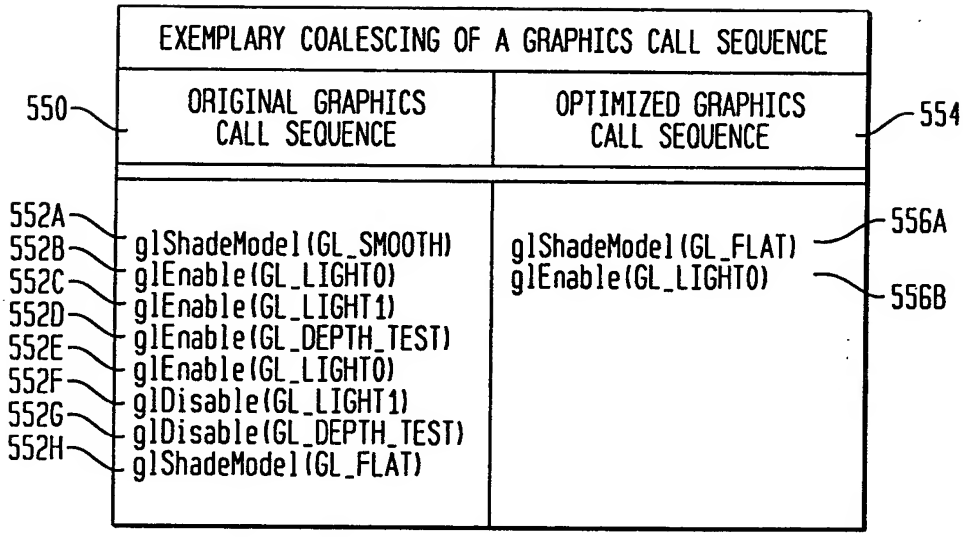

EXEMPLARY COALESCING OF A GRAPHICS CALL SEQUENCE
550
ORIGINAL GRAPHICS CALL SEQUENCE
OPTIMIZED GRAPHICS CALL SEQUENCE
554
552A glShadeModel(GL_SMOOTH)
552B glEnable(GL_LIGHT0)
552C glEnable(GL_LIGHT1)
552D glEnable(GL_DEPTH_TEST)
552E glEnable(GL_LIGHT0)
552F glDisable(GL_LIGHT1)
552G glDisable(GL_DEPTH_TEST)
552H glShadeModel(GL_FLAT)
glShadeModel(GL_FLAT)
556A
glEnable(GL_LIGHT0)
556B

FIG. 5D

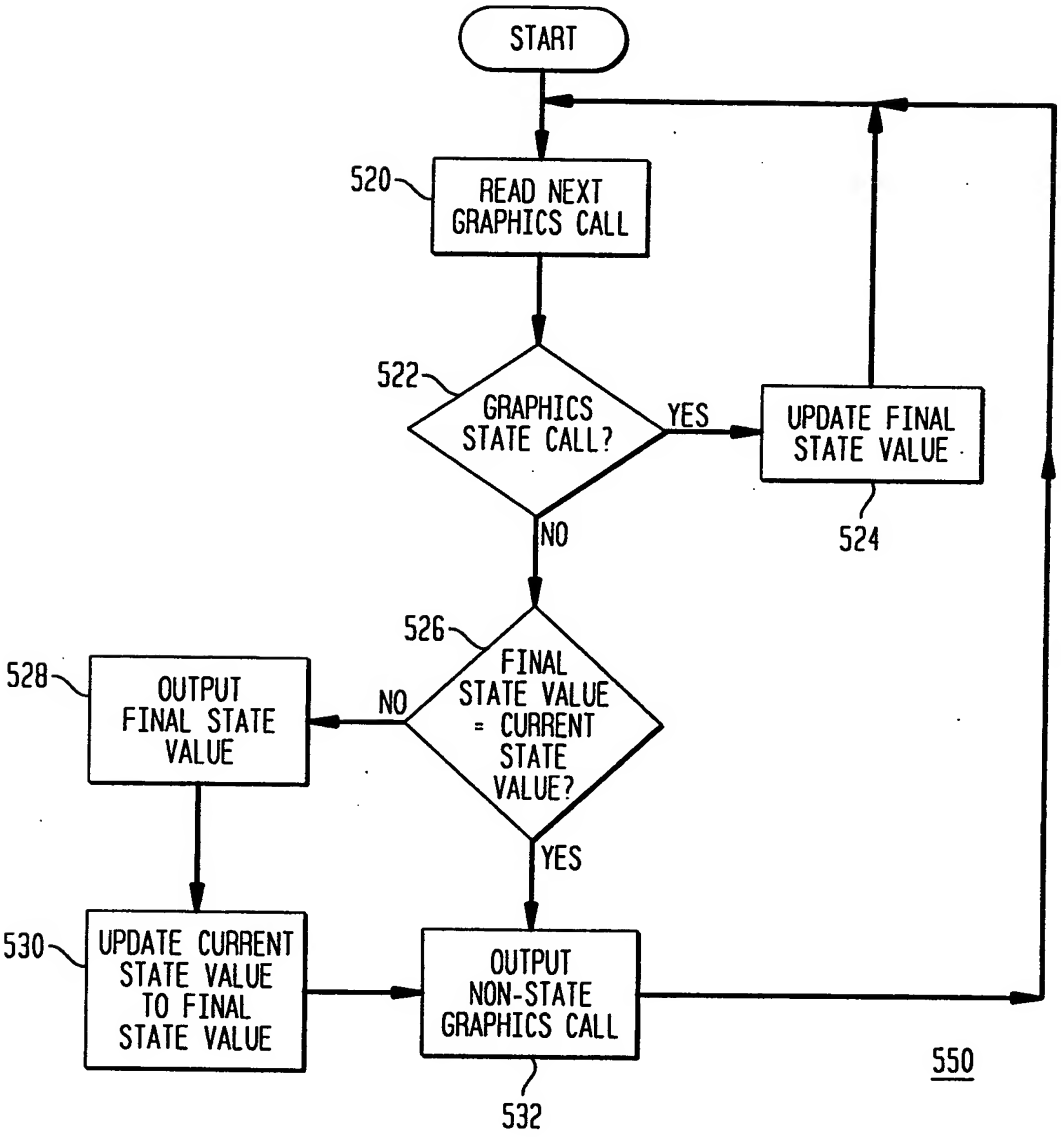

START
520
READ NEXT GRAPHICS CALL
522
GRAPHICS STATE CALL?
YES
UPDATE FINAL STATE VALUE
524
NO
526
FINAL STATE VALUE = CURRENT STATE VALUE?
NO
528
OUTPUT FINAL STATE VALUE
YES
530
UPDATE CURRENT STATE VALUE TO FINAL STATE VALUE
OUTPUT NON-STATE GRAPHICS CALL
532
550

FIG. 6A

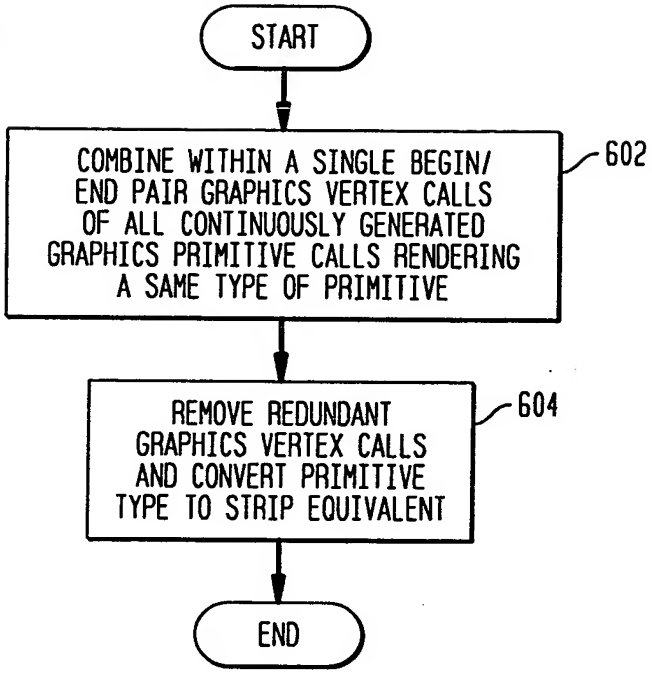

START
COMBINE WITHIN A SINGLE BEGIN/ END PAIR GRAPHICS VERTEX CALLS OF ALL CONTINUOUSLY GENERATED GRAPHICS PRIMITIVE CALLS RENDERING A SAME TYPE OF PRIMITIVE
602
REMOVE REDUNDANT GRAPHICS VERTEX CALLS AND CONVERT PRIMITIVE TYPE TO STRIP EQUIVALENT
604
END

FIG. 6B

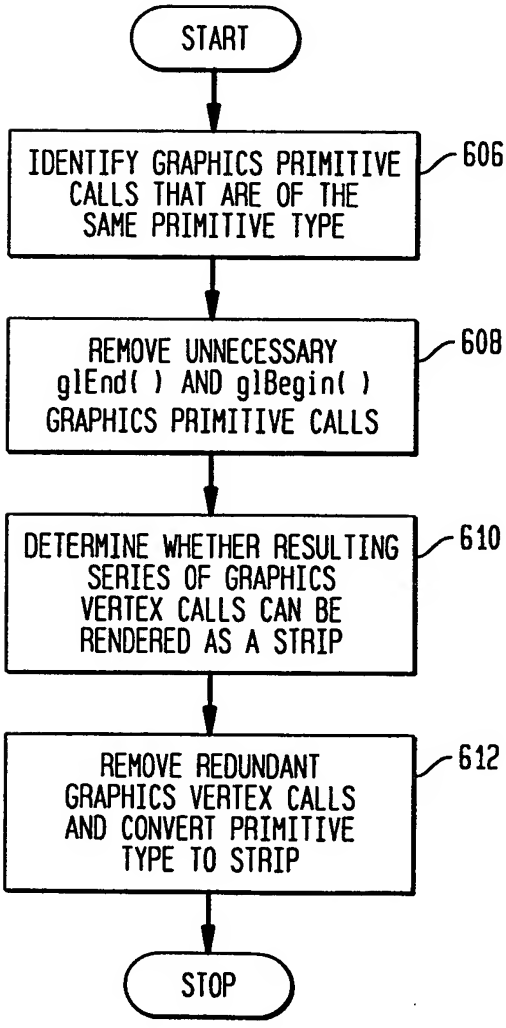

START
IDENTIFY GRAPHICS PRIMITIVE CALLS THAT ARE OF THE SAME PRIMITIVE TYPE
606
REMOVE UNNECESSARY glEnd( ) AND glBegin( ) GRAPHICS PRIMITIVE CALLS
608
DETERMINE WHETHER RESULTING SERIES OF GRAPHICS VERTEX CALLS CAN BE RENDERED AS A STRIP
610
REMOVE REDUNDANT GRAPHICS VERTEX CALLS AND CONVERT PRIMITIVE TYPE TO STRIP
612
STOP

FIG. 6C

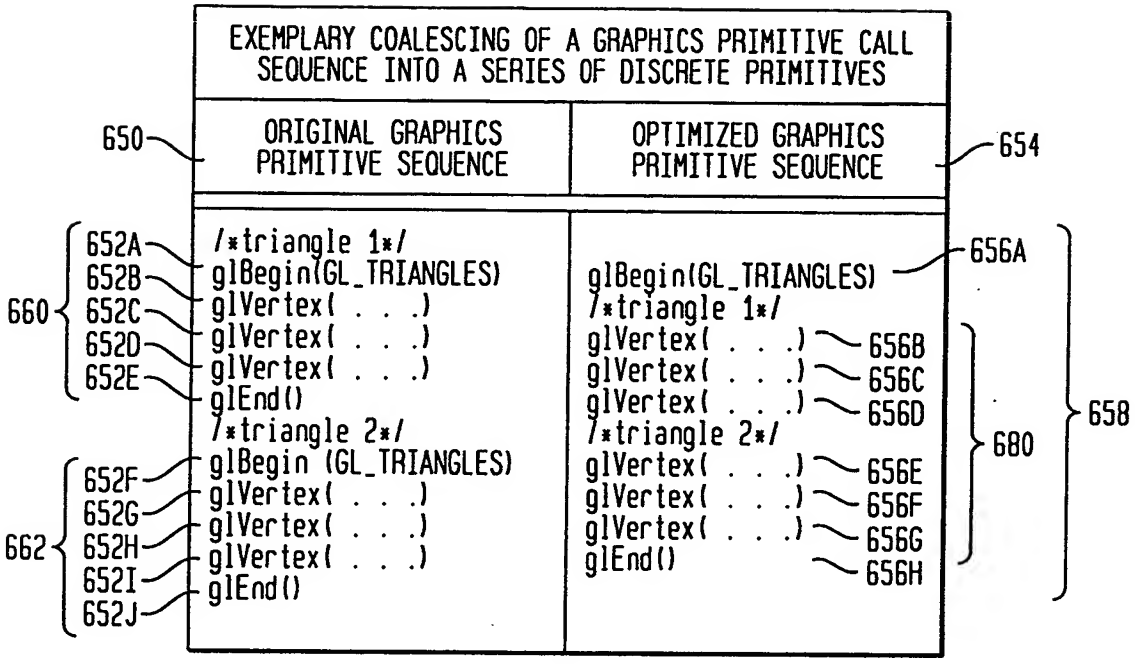

EXEMPLARY COALESCING OF A GRAPHICS PRIMITIVE CALL SEQUENCE INTO A SERIES OF DISCRETE PRIMITIVES
650
ORIGINAL GRAPHICS PRIMITIVE SEQUENCE
OPTIMIZED GRAPHICS PRIMITIVE SEQUENCE
654
652A
652B
652C
652D
652E
660
/*triangle 1*/
glBegin(GL_TRIANGLES)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
/*triangle 2*/
glBegin (GL_TRIANGLES)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
652F
652G
652H
652I
652J
662
656A
glBegin(GL_TRIANGLES)
/*triangle 1*/
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
/*triangle 2*/
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
656B
656C
656D
656E
656F
656G
656H
680
658

FIG. 6D

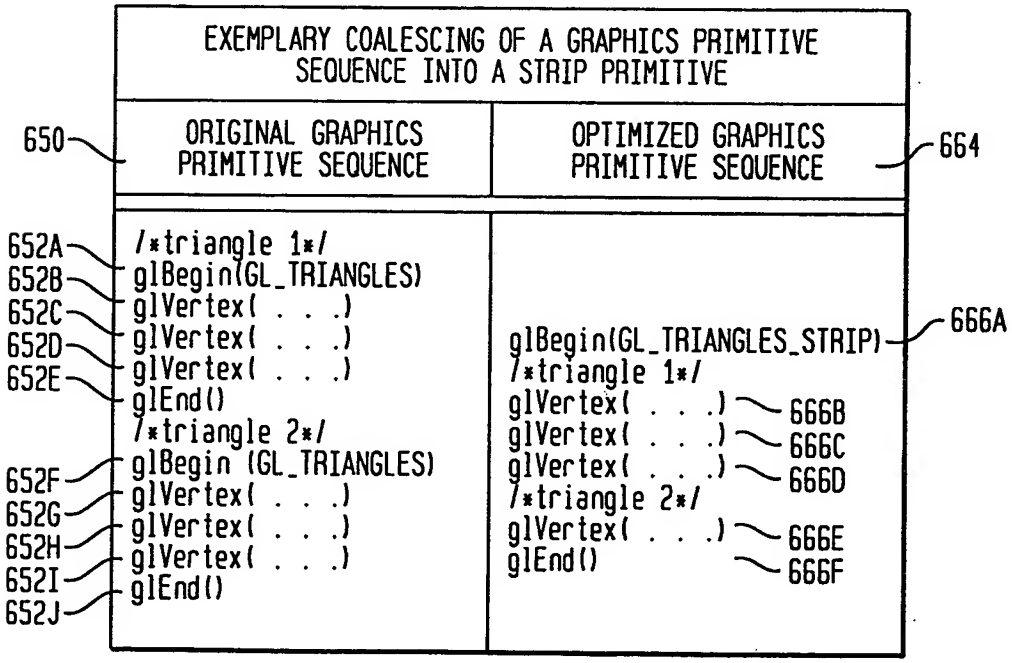

EXEMPLARY COALESCING OF A GRAPHICS PRIMITIVE SEQUENCE INTO A STRIP PRIMITIVE
650
ORIGINAL GRAPHICS PRIMITIVE SEQUENCE
OPTIMIZED GRAPHICS PRIMITIVE SEQUENCE
664
652A
652B
652C
652D
652E
/*triangle 1*/
glBegin(GL_TRIANGLES)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
/*triangle 2*/
652F
652G
652H
652I
652J
glBegin (GL_TRIANGLES)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
666A
glBegin(GL_TRIANGLES_STRIP)
/*triangle 1*/
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
/*triangle 2*/
glVertex(. . .)
glEnd()
666B
666C
666D
666E
666F

FIG. 6E

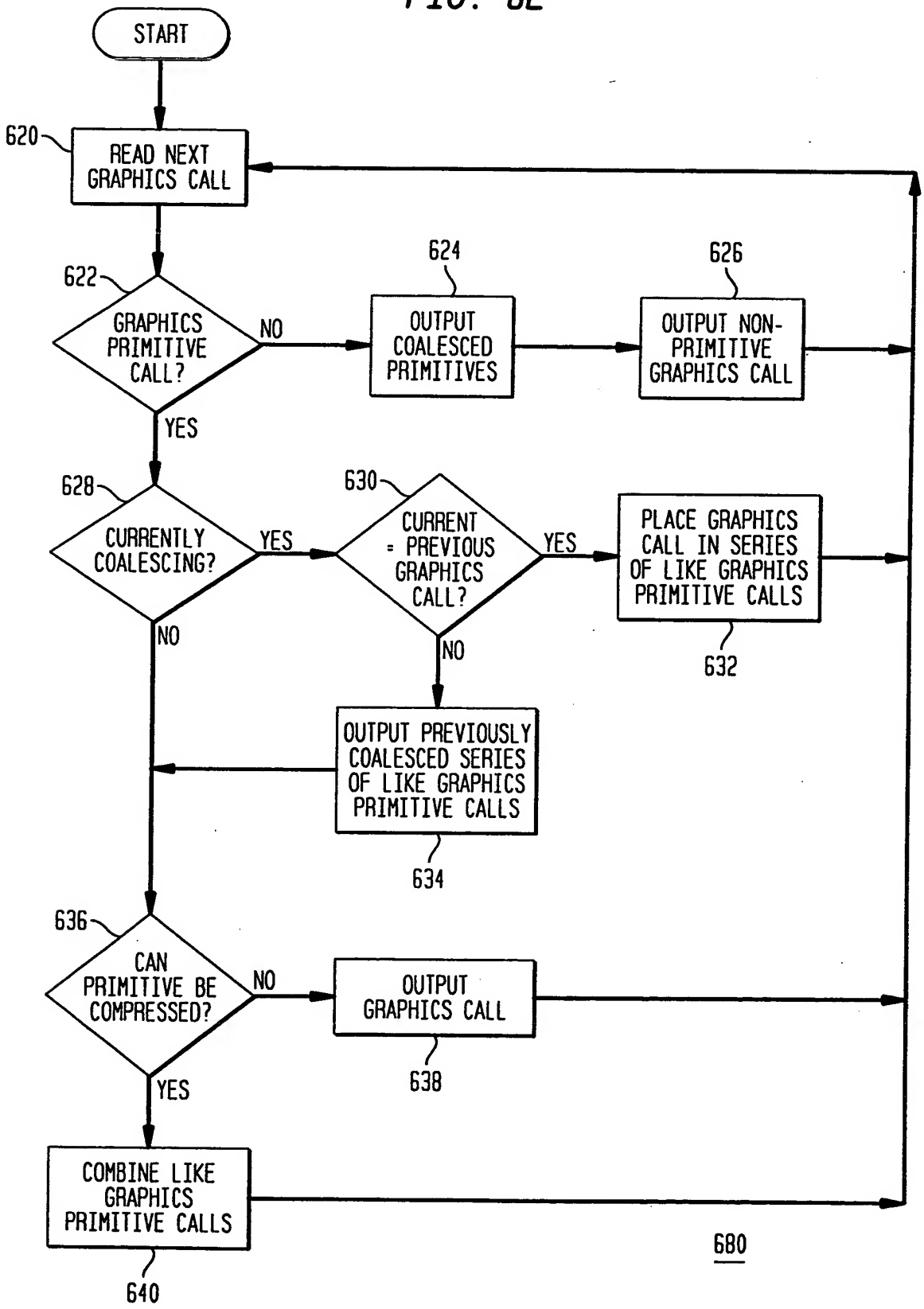

START
620
READ NEXT GRAPHICS CALL
622
GRAPHICS PRIMITIVE CALL?
NO
624
OUTPUT COALESCED PRIMITIVES
626
OUTPUT NON-PRIMITIVE GRAPHICS CALL
YES
628
CURRENTLY COALESCING?
YES
630
CURRENT = PREVIOUS GRAPHICS CALL?
YES
PLACE GRAPHICS CALL IN SERIES OF LIKE GRAPHICS PRIMITIVE CALLS
632
NO
NO
OUTPUT PREVIOUSLY COALESCED SERIES OF LIKE GRAPHICS PRIMITIVE CALLS
634
636
CAN PRIMITIVE BE COMPRESSED?
NO
OUTPUT GRAPHICS CALL
638
YES
COMBINE LIKE GRAPHICS PRIMITIVE CALLS
640
680

FIG. 7A

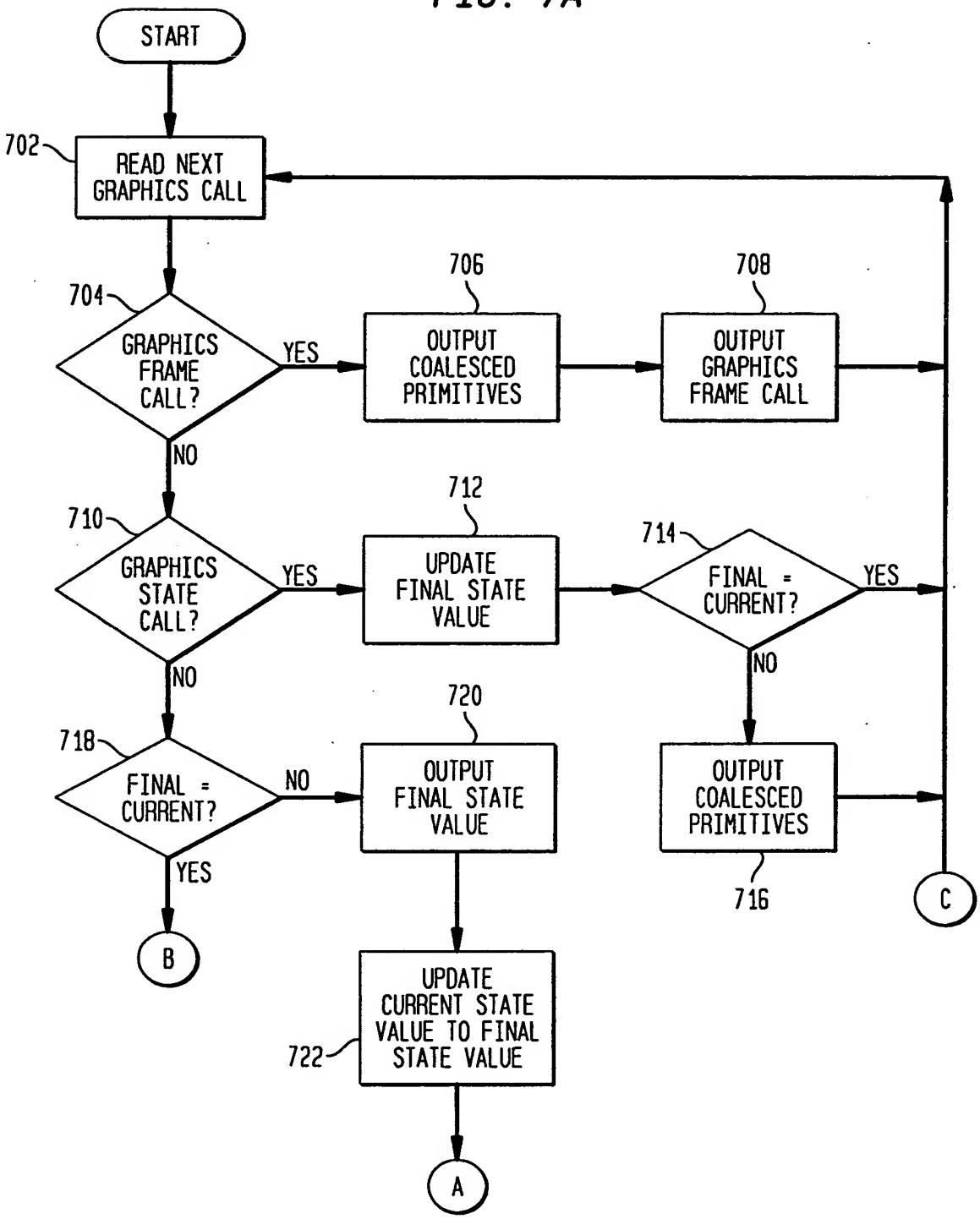

START
702
READ NEXT GRAPHICS CALL
704
GRAPHICS FRAME CALL?
YES
706
OUTPUT COALESCED PRIMITIVES
708
OUTPUT GRAPHICS FRAME CALL
NO
710
GRAPHICS STATE CALL?
YES
712
UPDATE FINAL STATE VALUE
714
FINAL = CURRENT?
YES
NO
716
OUTPUT COALESCED PRIMITIVES
C
NO
718
FINAL = CURRENT?
NO
720
OUTPUT FINAL STATE VALUE
YES
B
722
UPDATE CURRENT STATE VALUE TO FINAL STATE VALUE
A

FIG. 7B

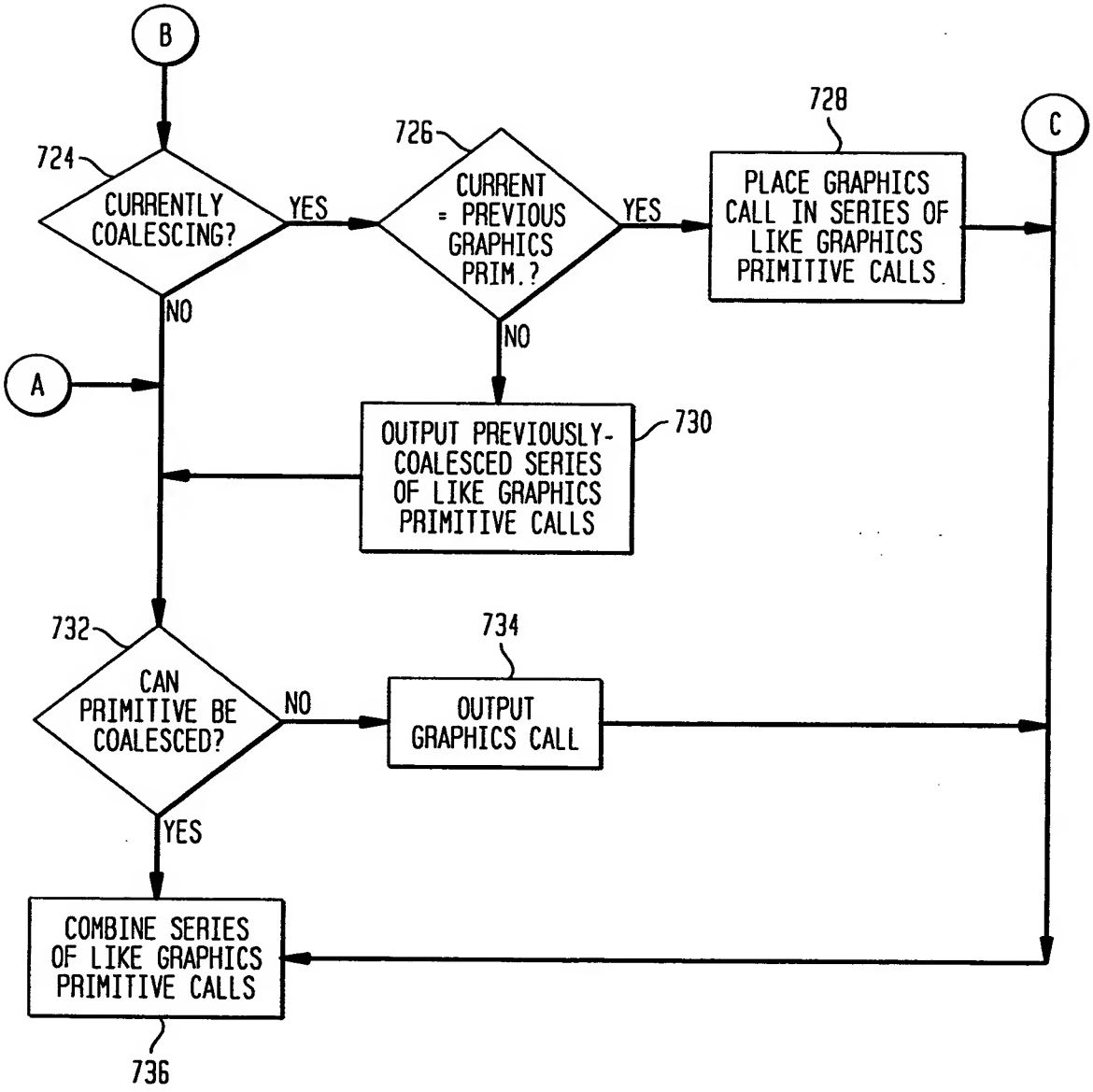

B
724
CURRENTLY COALESCING?
YES
NO
726
CURRENT = PREVIOUS GRAPHICS PRIM.?
YES
NO
728
PLACE GRAPHICS CALL IN SERIES OF LIKE GRAPHICS PRIMITIVE CALLS
C
A
OUTPUT PREVIOUSLY-COALESCED SERIES OF LIKE GRAPHICS PRIMITIVE CALLS
730
732
CAN PRIMITIVE BE COALESCED?
NO
YES
734
OUTPUT GRAPHICS CALL
COMBINE SERIES OF LIKE GRAPHICS PRIMITIVE CALLS
736

FIG. 8A

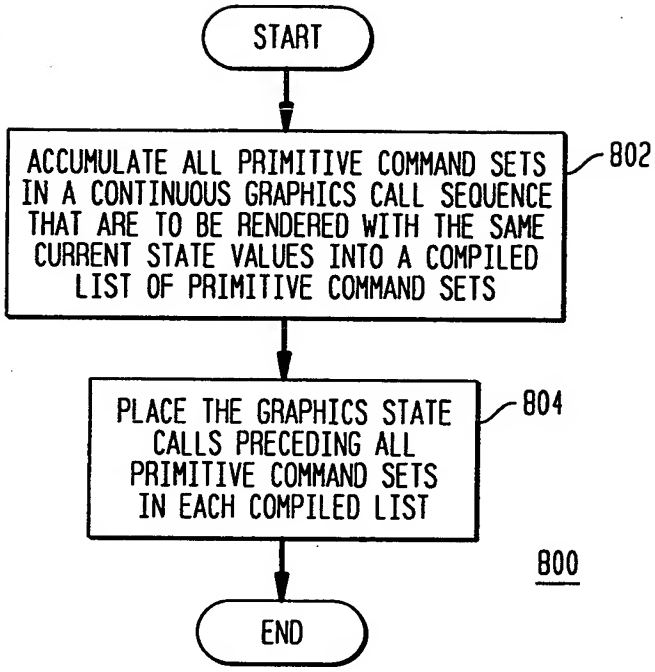

START
ACCUMULATE ALL PRIMITIVE COMMAND SETS IN A CONTINUOUS GRAPHICS CALL SEQUENCE THAT ARE TO BE RENDERED WITH THE SAME CURRENT STATE VALUES INTO A COMPILED LIST OF PRIMITIVE COMMAND SETS
802
PLACE THE GRAPHICS STATE CALLS PRECEDING ALL PRIMITIVE COMMAND SETS IN EACH COMPILED LIST
804
800
END

FIG. 8B

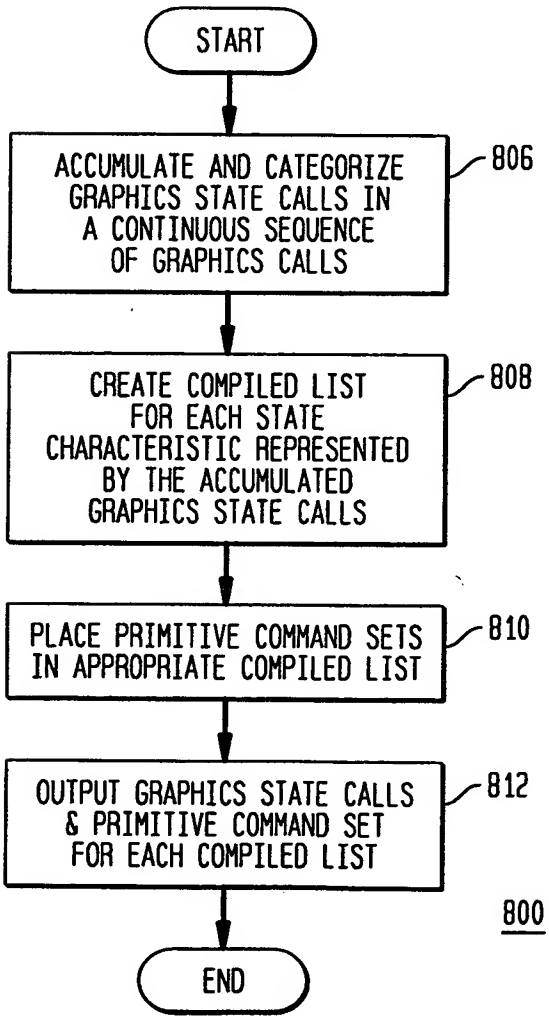

START
ACCUMULATE AND CATEGORIZE GRAPHICS STATE CALLS IN A CONTINUOUS SEQUENCE OF GRAPHICS CALLS
806
CREATE COMPILED LIST FOR EACH STATE CHARACTERISTIC REPRESENTED BY THE ACCUMULATED GRAPHICS STATE CALLS
808
PLACE PRIMITIVE COMMAND SETS IN APPROPRIATE COMPILED LIST
810
OUTPUT GRAPHICS STATE CALLS & PRIMITIVE COMMAND SET FOR EACH COMPILED LIST
812
800
END

FIG. 8C

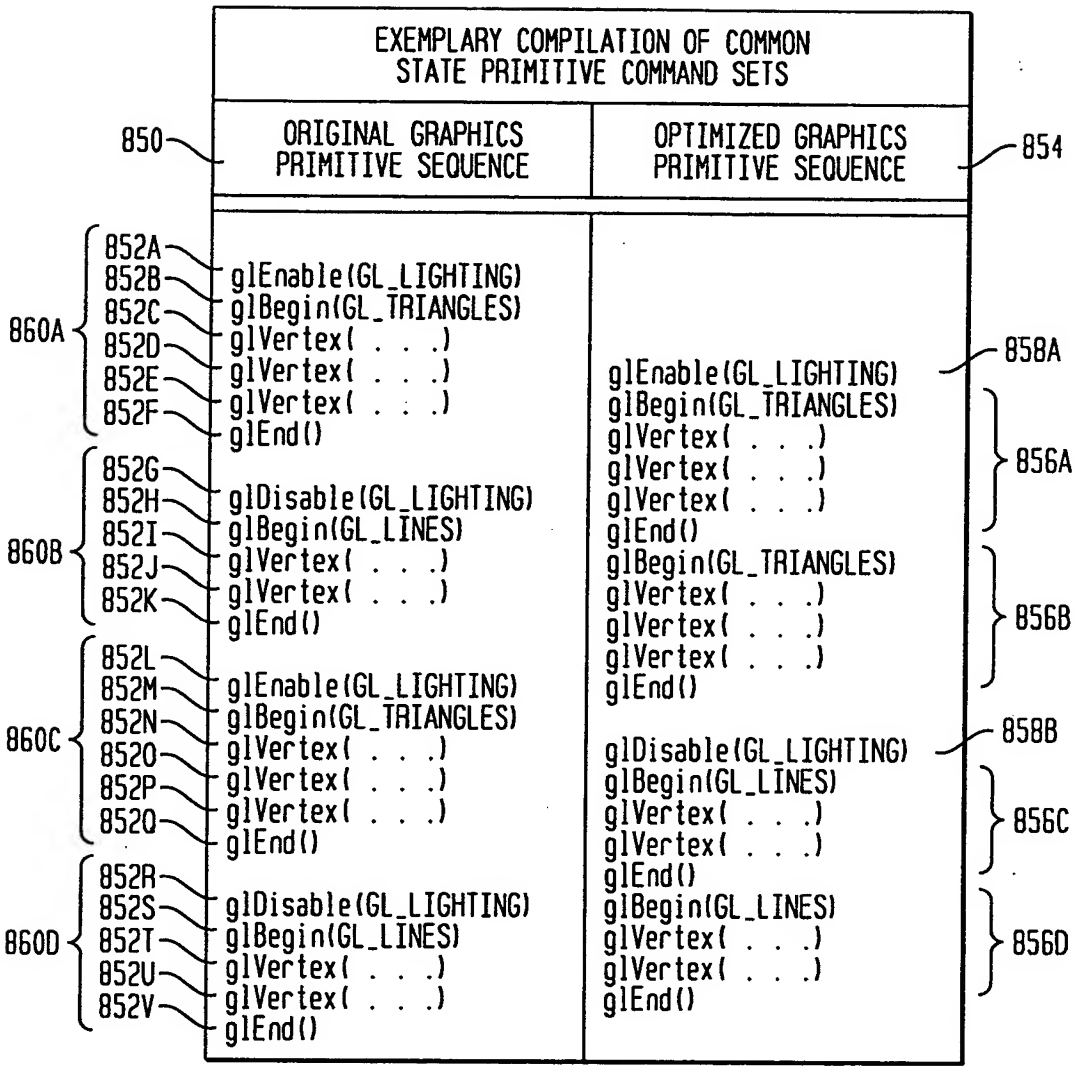
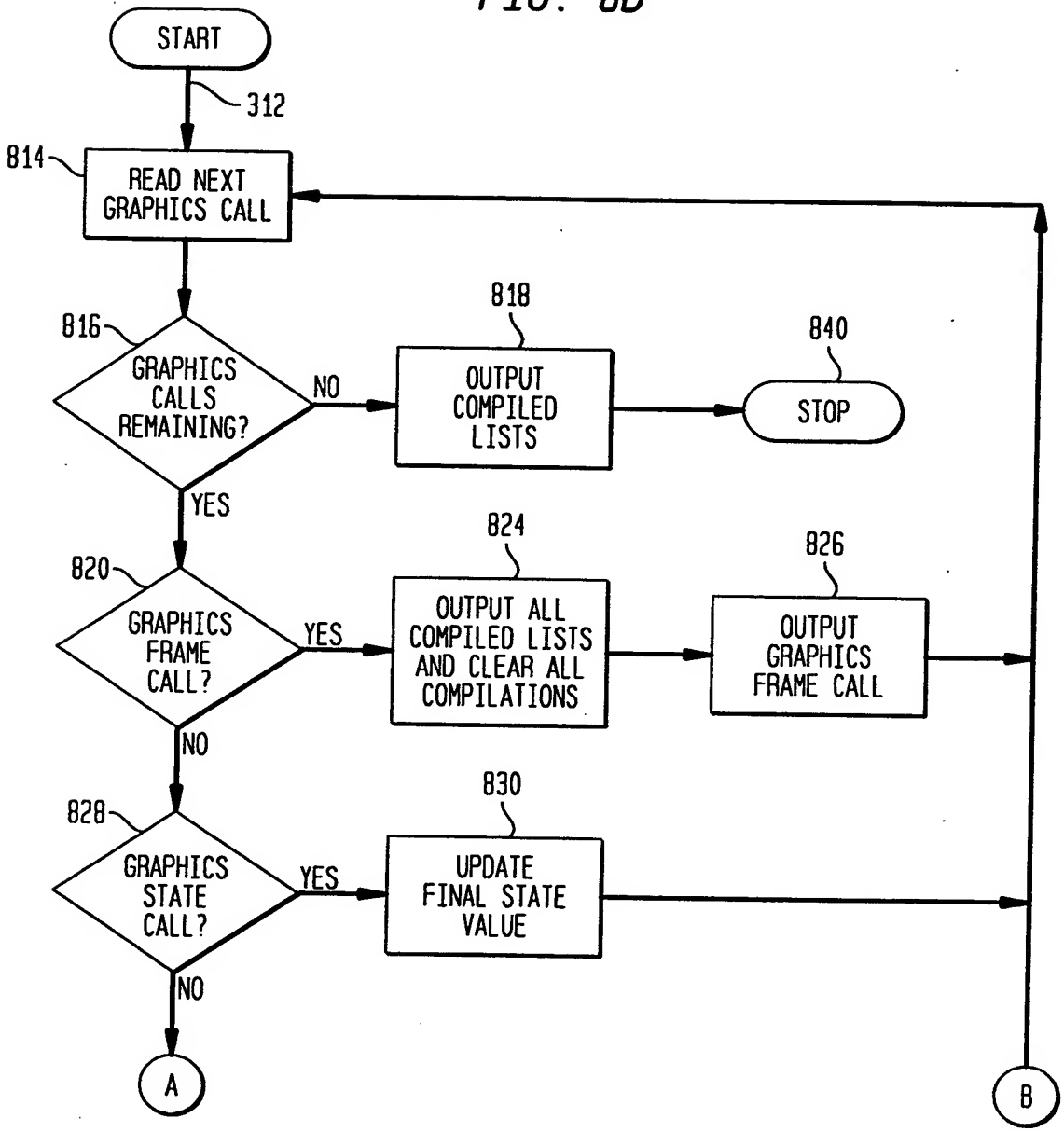

EXEMPLARY COMPILATION OF COMMON STATE PRIMITIVE COMMAND SETS
850
ORIGINAL GRAPHICS PRIMITIVE SEQUENCE
854
OPTIMIZED GRAPHICS PRIMITIVE SEQUENCE
852A glEnable(GL_LIGHTING)
852B glBegin(GL_TRIANGLES)
852C glVertex(. . .)
852D glVertex(. . .)
852E glVertex(. . .)
852F glEnd()
860A
852G glDisable(GL_LIGHTING)
852H glBegin(GL_LINES)
852I glVertex(. . .)
852J glVertex(. . .)
852K glEnd()
860B
852L glEnable(GL_LIGHTING)
852M glBegin(GL_TRIANGLES)
852N glVertex(. . .)
852O glVertex(. . .)
852P glVertex(. . .)
852Q glEnd()
860C
852R glDisable(GL_LIGHTING)
852S glBegin(GL_LINES)
852T glVertex(. . .)
852U glVertex(. . .)
852V glEnd()
860D
858A
glEnable(GL_LIGHTING)
glBegin(GL_TRIANGLES)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
856A
glBegin(GL_TRIANGLES)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
856B
858B
glDisable(GL_LIGHTING)
glBegin(GL_LINES)
glVertex(. . .)
glVertex(. . .)
glEnd()
856C
glBegin(GL_LINES)
glVertex(. . .)
glVertex(. . .)
glEnd()
856D

FIG. 8D


START
312
814
READ NEXT GRAPHICS CALL
816
GRAPHICS CALLS REMAINING?
NO
818
OUTPUT COMPILED LISTS
840
STOP
YES
820
GRAPHICS FRAME CALL?
YES
824
OUTPUT ALL COMPILED LISTS AND CLEAR ALL COMPILATIONS
826
OUTPUT GRAPHICS FRAME CALL
NO
828
GRAPHICS STATE CALL?
YES
830
UPDATE FINAL STATE VALUE
NO
A
B

880

FIG. 8E

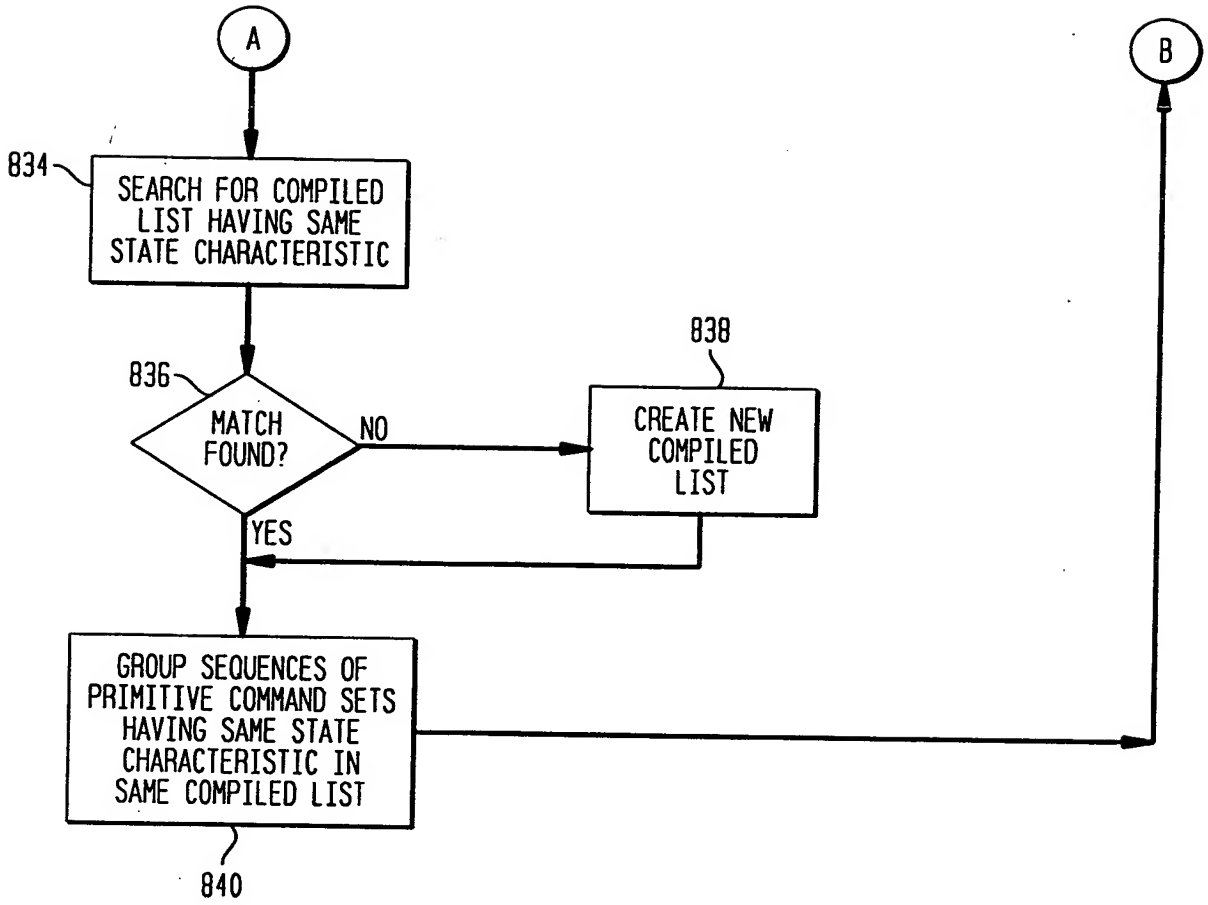

A
B
834
SEARCH FOR COMPILED LIST HAVING SAME STATE CHARACTERISTIC
836
MATCH FOUND?
NO
YES
838
CREATE NEW COMPILED LIST
GROUP SEQUENCES OF PRIMITIVE COMMAND SETS HAVING SAME STATE CHARACTERISTIC IN SAME COMPILED LIST
840

FIG. 9A

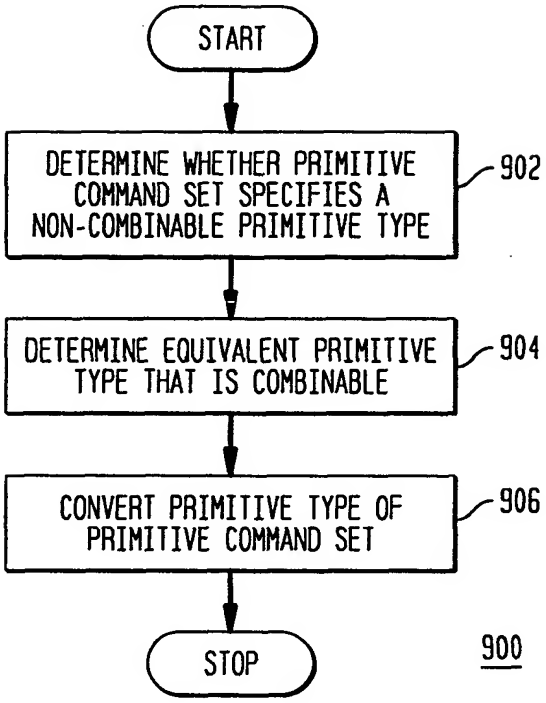

START
DETERMINE WHETHER PRIMITIVE COMMAND SET SPECIFIES A NON-COMBINABLE PRIMITIVE TYPE
902
DETERMINE EQUIVALENT PRIMITIVE TYPE THAT IS COMBINABLE
904
CONVERT PRIMITIVE TYPE OF PRIMITIVE COMMAND SET
906
STOP
900

FIG. 9B

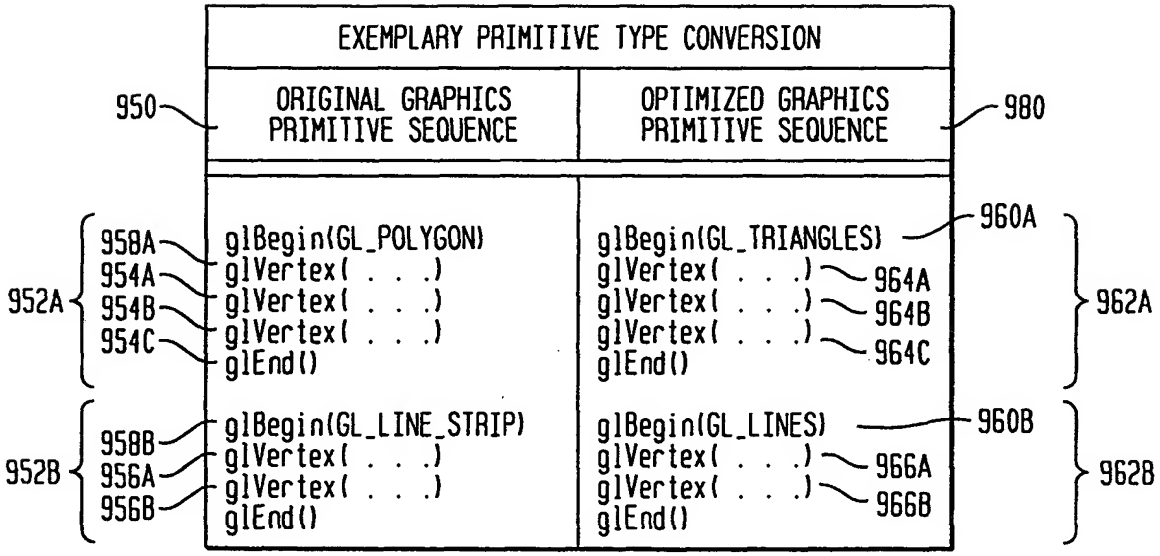

EXEMPLARY PRIMITIVE TYPE CONVERSION
950
ORIGINAL GRAPHICS PRIMITIVE SEQUENCE
OPTIMIZED GRAPHICS PRIMITIVE SEQUENCE
980
952A
958A glBegin(GL_POLYGON)
954A glVertex(. . .)
954B glVertex(. . .)
954C glVertex(. . .)
glEnd()
960A glBegin(GL_TRIANGLES)
glVertex(. . .) 964A
glVertex(. . .) 964B
glVertex(. . .) 964C
glEnd()
962A
952B
958B glBegin(GL_LINE_STRIP)
956A glVertex(. . .)
956B glVertex(. . .)
glEnd()
960B glBegin(GL_LINES)
glVertex(. . .) 966A
glVertex(. . .) 966B
glEnd()
962B

FIG. 9C

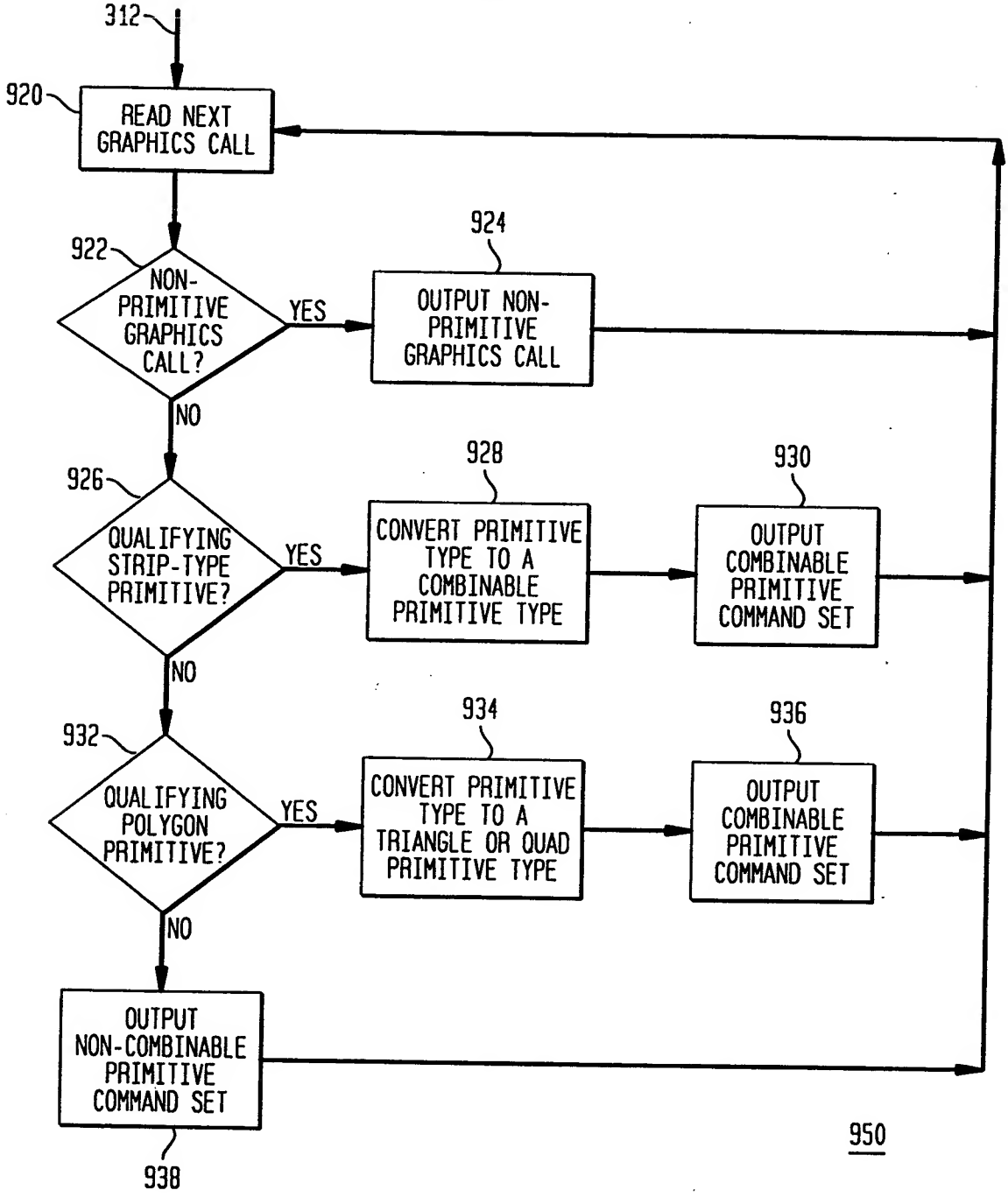

312
920
READ NEXT GRAPHICS CALL
922
NON-PRIMITIVE GRAPHICS CALL?
YES
924
OUTPUT NON-PRIMITIVE GRAPHICS CALL
NO
926
QUALIFYING STRIP-TYPE PRIMITIVE?
YES
928
CONVERT PRIMITIVE TYPE TO A COMBINABLE PRIMITIVE TYPE
930
OUTPUT COMBINABLE PRIMITIVE COMMAND SET
NO
932
QUALIFYING POLYGON PRIMITIVE?
YES
934
CONVERT PRIMITIVE TYPE TO A TRIANGLE OR QUAD PRIMITIVE TYPE
936
OUTPUT COMBINABLE PRIMITIVE COMMAND SET
NO
OUTPUT NON-COMBINABLE PRIMITIVE COMMAND SET
938
950

FIG. 10A

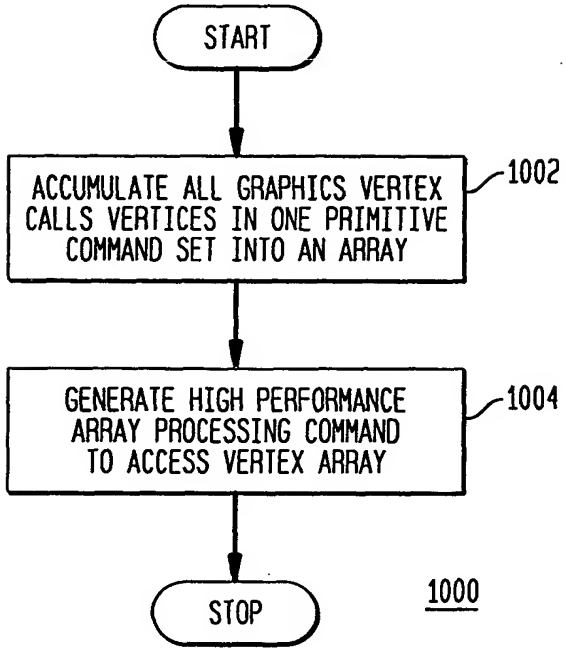


FIG. 10B

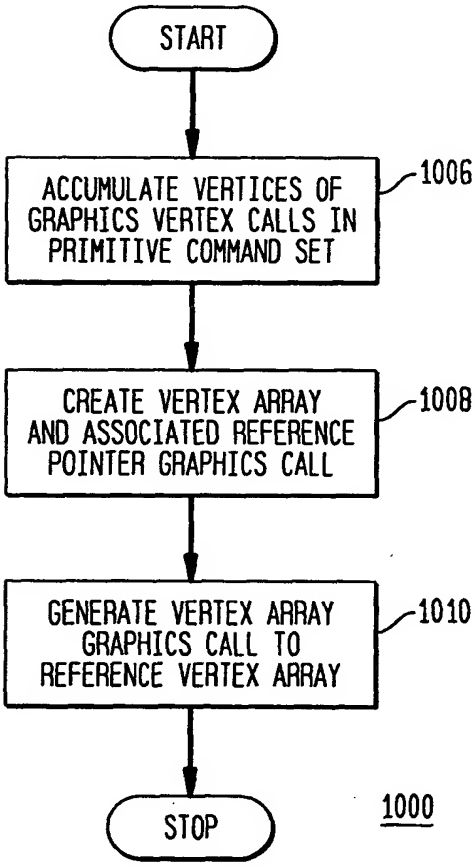


FIG. 10C

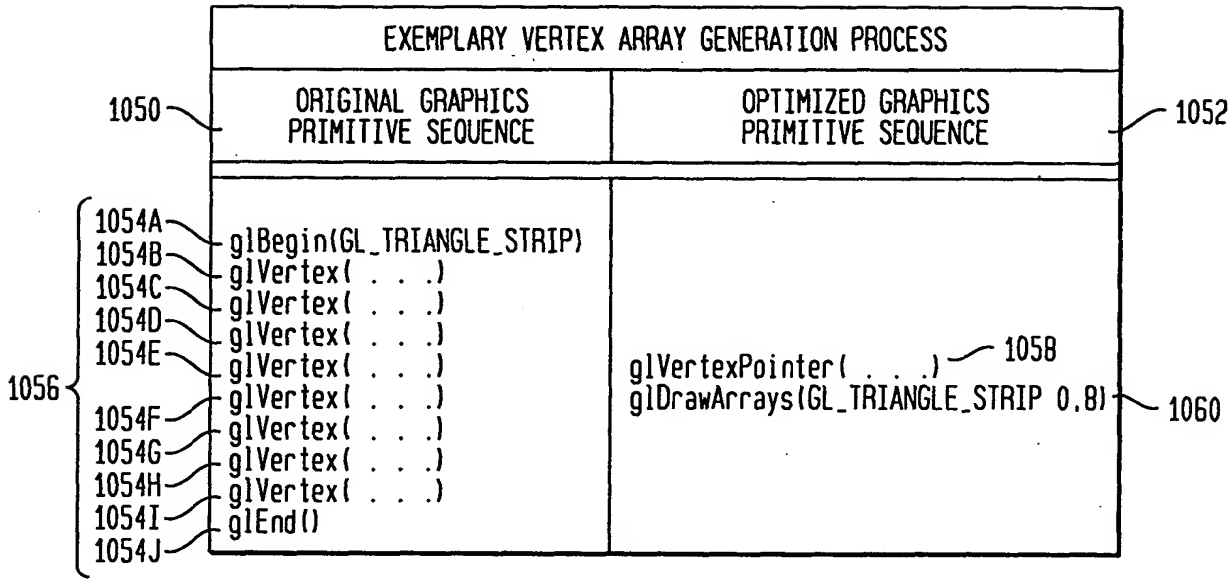

EXEMPLARY VERTEX ARRAY GENERATION PROCESS
1050
ORIGINAL GRAPHICS PRIMITIVE SEQUENCE
OPTIMIZED GRAPHICS PRIMITIVE SEQUENCE
1052
1054A
1054B
1054C
1054D
1054E
1054F
1054G
1054H
1054I
1054J
1056
glBegin(GL_TRIANGLE_STRIP)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glVertex(. . .)
glEnd()
glVertexPointer(. . .)
1058
glDrawArrays(GL_TRIANGLE_STRIP 0,8)
1060

FIG. 10D

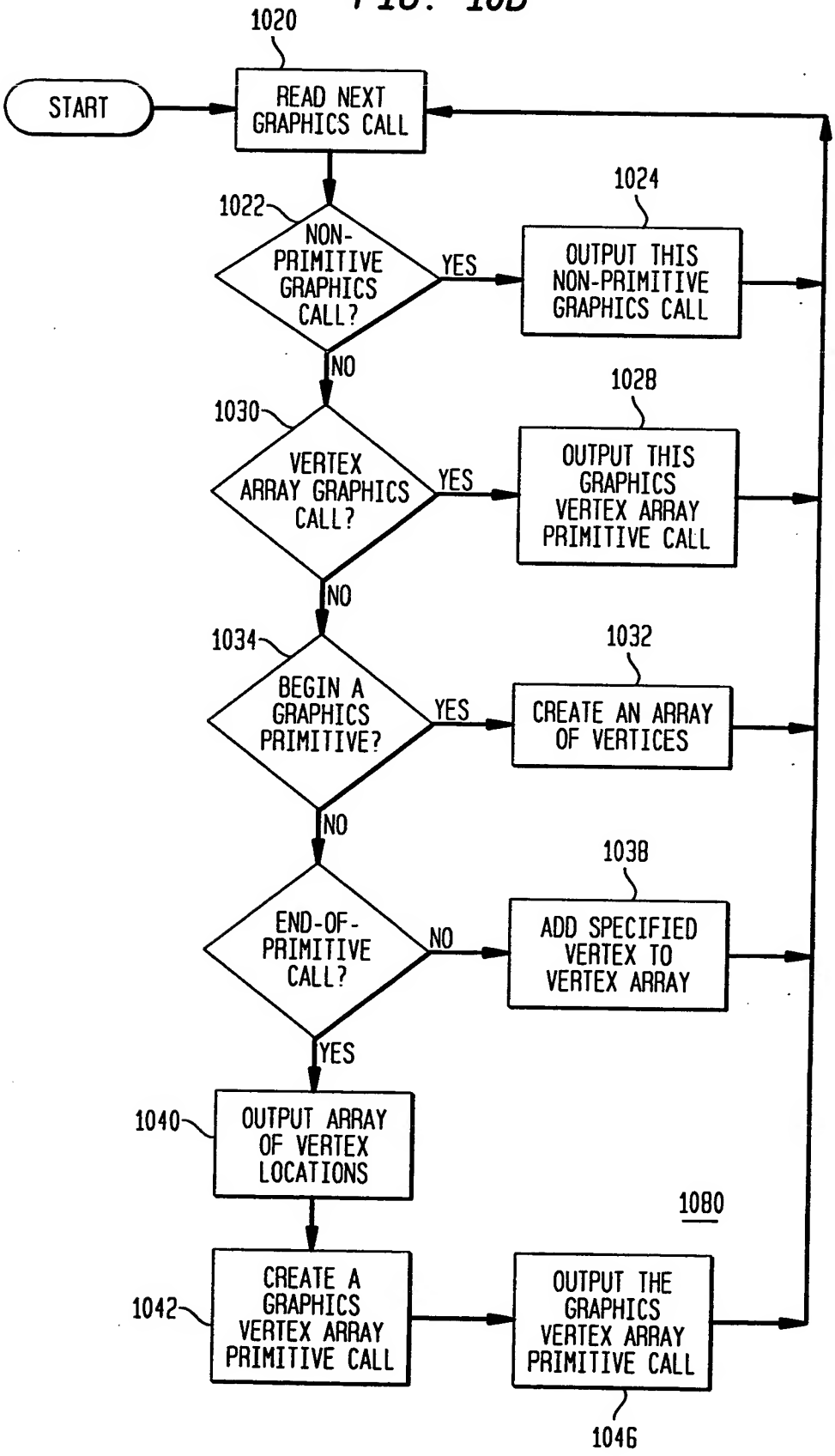

1020
START
READ NEXT GRAPHICS CALL
1022
NON-PRIMITIVE GRAPHICS CALL?
YES
1024
OUTPUT THIS NON-PRIMITIVE GRAPHICS CALL
NO
1030
VERTEX ARRAY GRAPHICS CALL?
YES
1028
OUTPUT THIS GRAPHICS VERTEX ARRAY PRIMITIVE CALL
NO
1034
BEGIN A GRAPHICS PRIMITIVE?
YES
1032
CREATE AN ARRAY OF VERTICES
NO
END-OF-PRIMITIVE CALL?
NO
1038
ADD SPECIFIED VERTEX TO VERTEX ARRAY
YES
1040
OUTPUT ARRAY OF VERTEX LOCATIONS
1080
1042
CREATE A GRAPHICS VERTEX ARRAY PRIMITIVE CALL
OUTPUT THE GRAPHICS VERTEX ARRAY PRIMITIVE CALL
1046

FIG. 11A

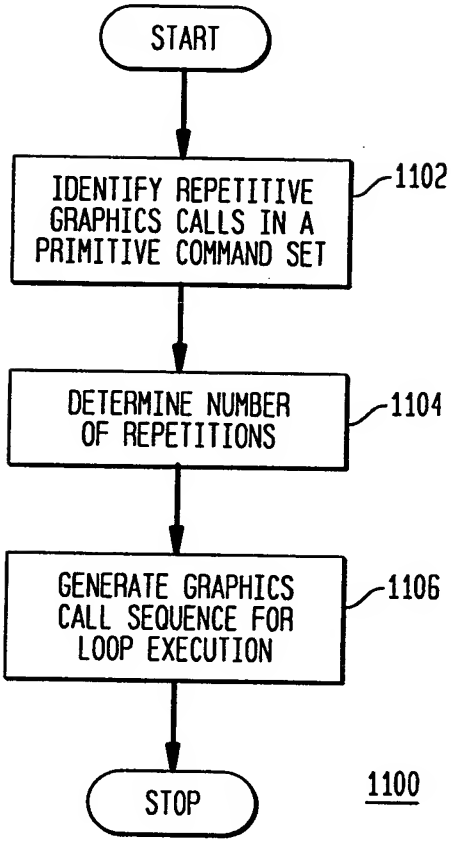

START
IDENTIFY REPETITIVE GRAPHICS CALLS IN A PRIMITIVE COMMAND SET
1102
DETERMINE NUMBER OF REPETITIONS
1104
GENERATE GRAPHICS CALL SEQUENCE FOR LOOP EXECUTION
1106
STOP
1100

FIG. 11B

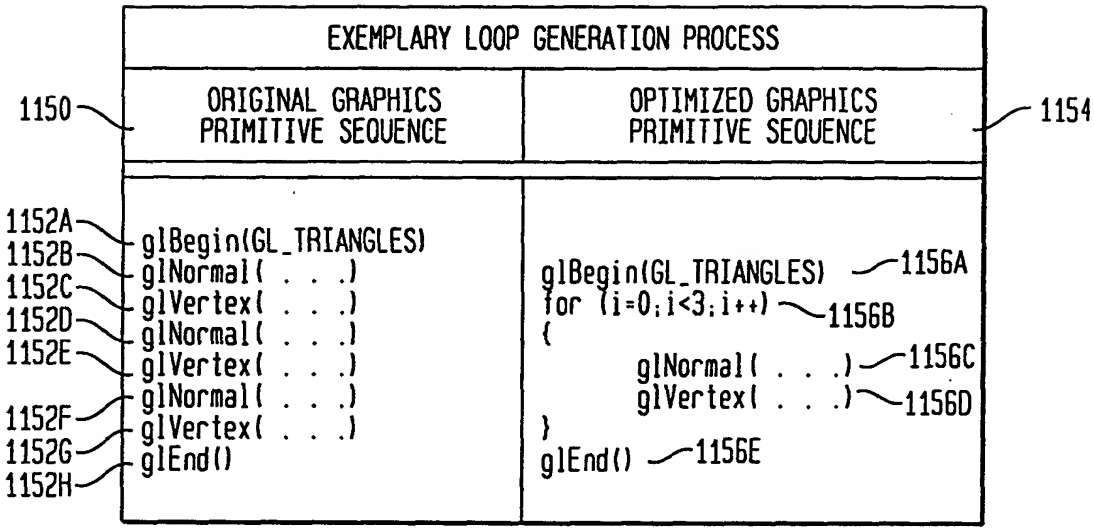
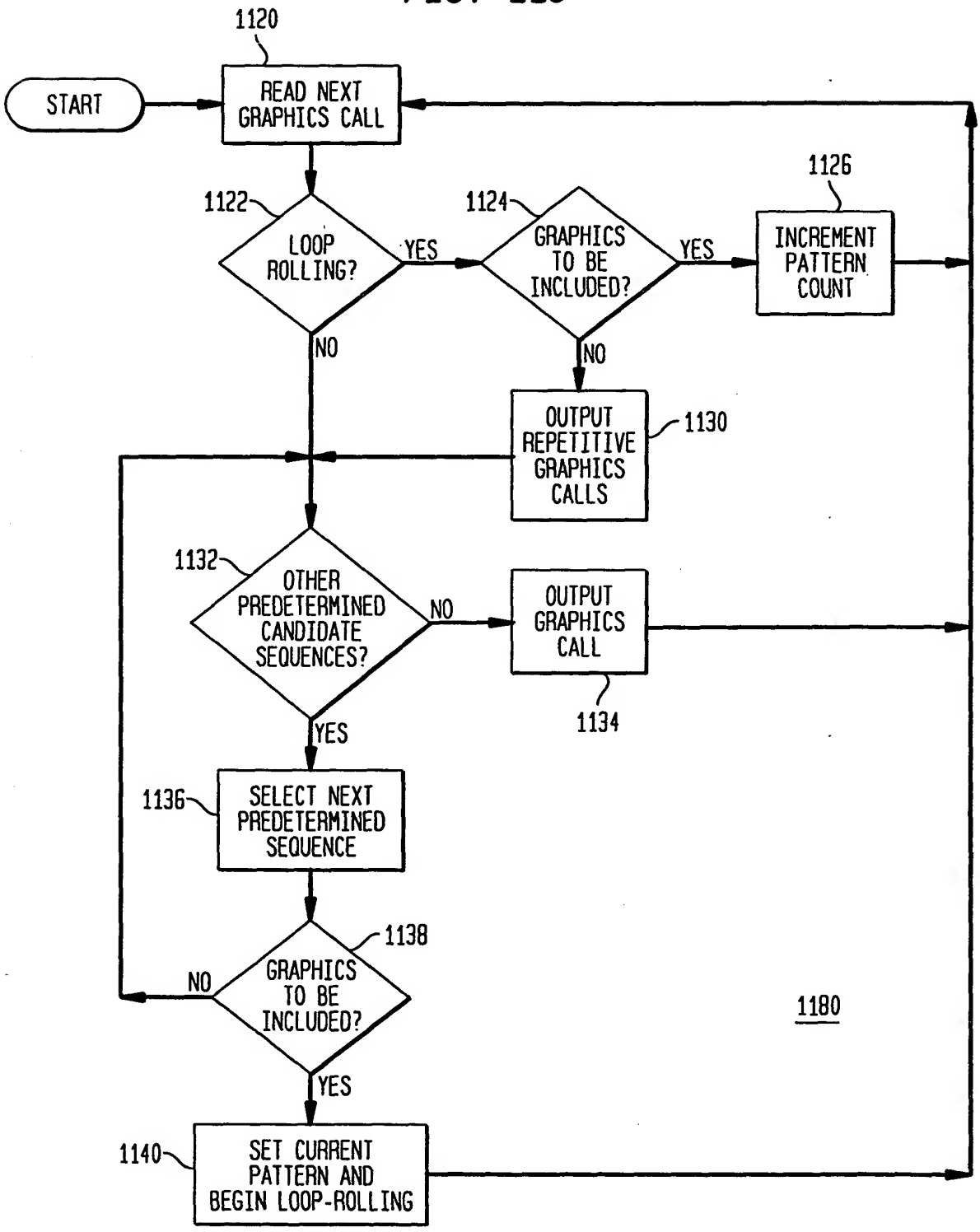

EXEMPLARY LOOP GENERATION PROCESS
1150
ORIGINAL GRAPHICS PRIMITIVE SEQUENCE
OPTIMIZED GRAPHICS PRIMITIVE SEQUENCE
1154
1152A
1152B
1152C
1152D
1152E
1152F
1152G
1152H
glBegin(GL_TRIANGLES)
glNormal(. . .)
glVertex(. . .)
glNormal(. . .)
glVertex(. . .)
glNormal(. . .)
glVertex(. . .)
glEnd()
glBegin(GL_TRIANGLES)
1156A
for (i=0;i<3;i++)
1156B
{
glNormal(. . .)
1156C
glVertex(. . .)
1156D
}
glEnd()
1156E

FIG. 11C


1120
START
READ NEXT GRAPHICS CALL
1122
LOOP ROLLING?
YES
1124
GRAPHICS TO BE INCLUDED?
YES
1126
INCREMENT PATTERN COUNT
NO
NO
OUTPUT REPETITIVE GRAPHICS CALLS
1130
1132
OTHER PREDETERMINED CANDIDATE SEQUENCES?
NO
OUTPUT GRAPHICS CALL
1134
YES
1136
SELECT NEXT PREDETERMINED SEQUENCE
1138
NO
GRAPHICS TO BE INCLUDED?
1180
YES
1140
SET CURRENT PATTERN AND BEGIN LOOP-ROLLING